What is claimed is:

[Claim 1] (currently amended) <u>A system for evaluating a set of rules, the system</u> <u>comprising; a rule engine implementing the RETE algorithm with novel extensions capable</u> <u>of distributing patterns to one or more systems, a set of rules defined by an user, and a set</u> <u>of data provided by the user</u> A rule engine which implements the RETE algorithm with novel extensions, which support distributed pattern matching.

[Claim 2] (currently amended) A system according to claim 1 <u>wherein the system</u> converts an object structure <u>to an internal structure, which the rule engine uses to perform</u> <u>operations and reason over data</u> into un-ordered facts within the engine.

[Claim 3] (currently amended) A system according to claim 1 <u>wherein the system stores</u> <u>data in a database called working memory; which may span multiple partitions across</u> <u>several computer systems and is capable of distributing indexes across multiple rule</u> <u>engines</u> implements the notion of working memory as described in reference article 1.

[Claim 4] (currently amended) <u>A system according to claim 1 wherein the system creates</u> <u>input nodes for each class type and is capable of propagating data through a local network,</u> <u>or remotely to a rule engine running on a separate computer system</u> Input nodes as defined in reference article 1 are extended with the capability to route objects to a remote rule engine or use it locally.

[Claim 5] (currently amended) <u>A system according to claim 1 wherein the system uses a 2</u> <u>input Join node to evaluate 2 or more data elements and determines if the data should</u> <u>propagate further through the network on a separate system or locally</u> Join nodes as defined in reference article 1 are extended with the capability to route matched patterns locally or remotely.

[Claim 6] (currently amended) <u>A system according to claim 1 wherein the system uses a</u> <u>Terminal nodes to indicate all the conditions of a rule have been met and is a candidate for</u>

immediate or delayed execution locally or on a remote computer system ~~Terminal nodes as defined in reference article 1 are extended with the capability to route matched patterns locally or remotely~~.

[Claim 7] (currently amended) A <u>pattern for a given class type</u> ~~template~~ according to claim 2 uses a linear sequence of nodes representing the patterns to match for the class <u>type; these nodes are called intra-element nodes</u> ~~an object~~.

[Claim 8] (currently amended) A system according to claim 1 <u>wherein the system</u> implements <u>a component</u> ~~an abstraction layer~~ for retrieving remote facts <u>residing in a separate rule engine process</u>.

[Claim 9] (currently amended) A system according to claim 1 <u>wherein the system</u> uses a call back mechanism between the working memory, the rule engine and the objects; which objects use to notify the rule engine of changes.

[Claim 10] (currently amended) A system according to claim 1 <u>wherein the system</u> requires object instantiations to implement a ~~base~~ <u>defined</u> interface for the call back mechanism, <u>which may be added to the classes using byte code manipulation</u>.

[Claim 11] (currently amended) A system according to claim 1 <u>wherein the system</u> monitors the resource usage; <u>which is used to determine when RETE nodes are distribute to other rule engines</u>.

[Claim 12] (currently amended) A system according to claim 1 <u>wherein the system</u> uses rules to manage the distribution of nodes to remote systems.

[Claim 13] (currently amended) A system according to claim 1 <u>wherein the system</u> distributes pattern matching by ~~serializing~~ <u>copying</u> the nodes <u>and sending them</u> to a remote system; <u>which are then added to remote engine's network</u>.

[Claim 14] (currently amended) A system according to claim 1 <u>wherein the system</u> distributes <u>the class type node</u> ~~input~~, <u>2 input node</u> ~~join~~, terminal <u>node</u> and intra-element nodes to a remote system.

[Claim 15] (original) Distributed nodes according to claim 14 maintains a list of remote systems which depend on the indexes produced by pattern matching.

[Claim 16] (currently amended) A system according to claim 1 wherein the system will ~~serialize~~ transmit the values of an object to a remote system, if the corresponding pattern matches against a remote object pattern, or it may perform join using the indexes directly.

[Claim 17] (currently amended) A system according to claim 1 wherein the system may serialize the data object and its values to a remote system, if the object contains procedural logic and functional attachments ~~including remote service method calls~~.

[Claim 18] (currently amended) A system according to claim 1 wherein the system may serialize the values of an object to a remote system and the receiving system may create a new instance of the object for pattern matching.

[Claim 19] (original) Objects according to claims 16 to 18 are considered temporal by the rule engine if the object's original instance and nodes reside on a remote system.

[Claim 20] (currently amended) A system according to claim 1 wherein the system defines three types of input channels: standard input, data distribution and node distribution.

[Claim 21] (currently amended) A system according to claim 1 wherein the system defines a set of application interfaces and components to handle incoming data input, events and requests for pattern matching.

[Claim 22] (original) Input according to claim 21 is defined as standard input channel.

[Claim 23] (currently amended) A system according to claim 1 wherein the system defines a data distribution channel for sending and receiving remote data between rule engines.

[Claim 24] (currently amended) A system according to claim 1 wherein the system defines a pattern distribution channel for distributing RETE nodes and patterns.

[Claim 25] (currently amended) A system according to claim 1 wherein the system considers an object as temporal if it was sent through the data distribution channels.

[Claim 26] (currently amended) Temporal objects according to claim 19 and 25 are used by the engine to perform pattern matching and ~~these objecsts are~~ may be discarded immediately after the pattern matching process is complete.

[Claim 27] (currently amended) A system according to claim 1 <u>wherein the system</u> will route the <u>index</u> results of claim 26 back to the originating system using the data distribution channel.

[Claim 28] (currently amended) A system according to claim 1 <u>wherein the system</u> will update the index of the join and terminal nodes as a result of pattern matching according to claim 26.

[Claim 29] (currently amended) A system according to claim 1 <u>wherein the system</u> processes the outputs of the rule and produces a set of results, if the original event began locally.

[Claim 30] (currently amended) A system according to claim 1 <u>wherein the system</u> uses messaging system to route new events to a cluster of rule engines.

[Claim 31] (original) A messaging system according to claim 30 filters new messages and routes them to the correct engine.

[Claim 32] (currently amended) A system according to claim 1 <u>wherein the system</u> uses messaging system to route the final results as defined by a rule to the recipient.

[Claim 33] (currently amended) A system according to claim 1 <u>wherein the system</u> contains a component responsible for communicating with the messaging system.

[Claim 34] (original) A messaging component according to claim 33 is responsible for processing inbound events and generating new messages for outbound publication.

[Claim 35] (currently amended) A system according to claim 1 <u>wherein the system</u> prefers to process new <u>data</u> events asynchronously using a messaging system<u>, but is not required to use asynchronous communication</u>.

[Claim 36] (original) Distributed nodes according to claim 14 contain information about the originating engine, a timestamp of when the nodes were distributed and a priority attribute.

[Claim 37] (original) The priority attribute according to claim 36 may be used by the engine to remove the nodes, if all local object instances have been removed from the working memory.

[Claim 38] (currently amended) Distributed nodes according to claim 14 will not be removed from the local pattern matching network, if data for those patterns is still being used, either in active rules about to fire or in remote rule engine instances procedural attachment calls.

[Claim 39] (currently amended) A system according to claim 1 wherein the system may forward node distribution messages, if the system does not have sufficient resources.

[Claim 40] (original) A forward message according to claim 19 must retain the location of the originating engine and add the current engine's unique runtime name to a list of recipients.

[Claim 41] (currently amended) A system according to claim 1 wherein the system will notify the producer of the node distribution message of success or failure.

[Claim 42] (original) Distributed nodes according to claim 14 may be distributed at deployment time.

[Claim 43] (currently amended) A system according to claim 1 wherein the system will set an attribute of the input node to indicate that the pattern has been distributed.

[Claim 44] (original) An input node according to claim 43 will maintain a list of the remote systems and the total number of data objects routed remotely.

[Claim 45] (currently amended) A system according to claim 1 wherein the system may not attempt to distribute nodes that were distributed by another rule engine. Instead, it

should notify the originating engine it cannot receive additional data until resources are available.

[Claim 46] (currently amended) A system according to claim 1 wherein the system may randomly select a remote engine to route data to, if the pattern is distributed to more than 1 engine.